

# Accurate 3D Action Recognition using Learning on the Grassmann Manifold

Rim Slama, Hazem Wannous, Mohamed Daoudi, Anuj Srivastava

► **To cite this version:**

Rim Slama, Hazem Wannous, Mohamed Daoudi, Anuj Srivastava. Accurate 3D Action Recognition using Learning on the Grassmann Manifold. Pattern Recognition, Elsevier, 2015, 48 (2), pp.556-567. hal-01056399

**HAL Id: hal-01056399**

**<https://hal.archives-ouvertes.fr/hal-01056399>**

Submitted on 20 Aug 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Accurate 3D Action Recognition using Learning on the Grassmann Manifold

Rim Slama<sup>a,b</sup>, Hazem Wannous<sup>a,b</sup>, Mohamed Daoudi<sup>b,c</sup>, Anuj Srivastava<sup>d</sup>

<sup>a</sup>*University Lille 1, Villeneuve d'Ascq, France*

<sup>b</sup>*LIFL Laboratory / UMR CNRS 8022, Villeneuve d'Ascq, France*

<sup>c</sup>*Institut Mines-Telecom / Telecom Lille, Villeneuve d'Ascq, France*

<sup>d</sup>*Florida State University, Department of Statistics, Tallahassee, USA*

---

## Abstract

In this paper we address the problem of modelling and analyzing human motion by focusing on 3D body skeletons. Particularly, our intent is to represent skeletal motion in a geometric and efficient way, leading to an accurate action-recognition system. Here an action is represented by a dynamical system whose observability matrix is characterized as an element of a Grassmann manifold. To formulate our learning algorithm, we propose two distinct ideas: (1) In the first one we perform classification using a Truncated Wrapped Gaussian model, one for each class in its own tangent space. (2) In the second one we propose a novel learning algorithm that uses a vector representation formed by concatenating local coordinates in tangent spaces associated with different classes and training a linear SVM. We evaluate our approaches on three public 3D action datasets: MSR-action 3D, UT-kinect and UCF-kinect datasets; these datasets represent different

---

*Email addresses:* [rim.slama@telecom-lille.fr](mailto:rim.slama@telecom-lille.fr) (Rim Slama),  
[hazem.wannous@telecom-lille.fr](mailto:hazem.wannous@telecom-lille.fr) (Hazem Wannous),  
[mohamed.daoudi@telecom-lille.fr](mailto:mohamed.daoudi@telecom-lille.fr) (Mohamed Daoudi), [anuj@stat.fsu.edu](mailto:anuj@stat.fsu.edu) (Anuj Srivastava)

kinds of challenges and together help provide an exhaustive evaluation. The results show that our approaches either match or exceed state-of-the-art performance reaching 91.21% on MSR-action 3D, 97.91% on UCF-kinect, and 88.5% on UT-kinect. Finally, we evaluate the latency, i.e. the ability to recognize an action before its termination, of our approach and demonstrate improvements relative to other published approaches.

*Keywords:* Human action recognition, Grassmann manifold, observational latency, depth images, skeleton, classification.

---

## 1. Introduction

Human action and activity recognition is one of the most active research topics in the computer vision community due to its many challenging issues. The motivation behind the great interest granted to action recognition is the large number of possible applications in consumer interactive entertainment and gaming [1], surveillance systems [2], life-care and home systems [3]. An extensive literature around this domain can be found in a number of fields including pattern recognition, machine learning, and human-machine interaction [4, 5].

The main challenges in action recognition systems are the accuracy of data acquisition and the dynamic modelling of the movements. The major problems, which can alter the way actions are perceived and consequently be recognized, are: occlusions, shadows and background extraction, lighting condition variations and viewpoint changes. The recent release of consumer depth cameras, like Microsoft Kinect, has significantly lighten these difficulties that reduce the action recognition performance in 2D video. These

17 cameras provide in addition to the RGB image a depth stream allowing to  
18 discern changes in depth in certain viewpoints.

19 More recently, Shotton et al. [6] have proposed a real-time approach for  
20 estimating 3D positions of body joints using extensive training on synthetic  
21 and real depth-streams. The accurate estimation obtained by such a low-  
22 cost acquisition depth sensor has provided new opportunities for human-  
23 computer-interaction applications, where popular gaming consoles involve  
24 the player directly in interaction with the computer. While these acquisition  
25 sensors and their accurate data are within everyone’s reach, the next research  
26 challenge is activity-driven.

27 In this paper we address the problem of modelling and analyzing human  
28 motion in the 3D human joint space. Particularly, our intent is to represent  
29 skeletal joint motion in a compact and efficient way that leads to an accurate  
30 action recognition. Our ultimate goal is to develop an approach that avoids  
31 an overly complex design of feature extraction and is able to recognize actions  
32 performed by different actors in different contexts.

33 Additionally, we study the ability of our approach for reducing latency:  
34 in other words, to quickly recognize human actions from the smallest number  
35 of frames possible to permit a reliable recognition of the action occurring.  
36 Furthermore, we analyze the impact of reducing the number of actions per  
37 class in the training set on the classifier’s accuracy.

38 In our approach, the spatio-temporal aspect of the action is considered  
39 and each movement is characterized by a structure incorporating the intrinsic  
40 nature of the data. We believe that 3D human joint motion data captures  
41 useful knowledge to understand the intrinsic motion structure, and a manifold

42 representation of such simple features can provide discriminating structure  
 43 for action recognition. This leads to manifold-based analysis, which has  
 44 been successfully used in many computer vision applications such as visual  
 45 tracking [7] and action recognition in 2D video [8, 9, 10, 11].

46 Our overall approach is sketched in Figure 1, which has the following  
 47 modules:

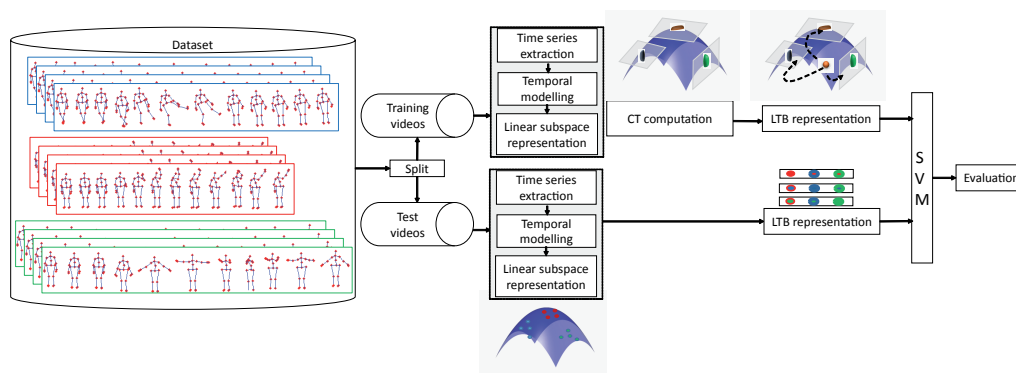


Figure 1: Overview of the approach. The illustrated pipeline is composed of two main modules: (1) temporal modelling of time series data and manifold representation (2) learning approach on the Control Tangent spaces on Grassman manifold, using Local Bundle Tangent representation of data.

48 First, given training videos recorded from depth camera, motion trajectories  
 49 from the 3D human joint in Euclidean space are extracted as time series.  
 50 Then, each motion represented by its time series is expressed as an autore-  
 51 gressive and moving average model (ARMA) in order to model its dynamic  
 52 process. The subspace spanned by columns of the observability matrix of  
 53 this model represents a point on a Grassmann manifold.

54 Second, using the Riemannian geometry of this manifold, we present a so-  
 55 lution for solving the classification problem. We studied statistical modelling

56 of inter- and intra-class variations in conjunction with appropriate tangent  
57 vectors on this manifold. While class samples are presented by a Grass-  
58 mann point cloud, we propose to learn Control Tangent (CT) spaces which  
59 represent the mean of each class.

60 Third, each observation of the learning process is projected on all CTs  
61 to form a Local Tangent Bundle (LTB) representation. This step allows  
62 obtaining a discriminative parameterization incorporating class separation  
63 properties and providing the input to a linear SVM classifier.

64 While given an unknown test video, to recognize its belonging to one of  
65  $N$  action classes, we apply the first step on the sequence to represent it as  
66 a point on the Grassmann manifold. Then, this point is presented by its  
67 LTB as done in learning step. In order to recognize the input action, SVM  
68 classifier is performed.

69 The rest of the paper is organized as follows: In section 1, the state-of-  
70 the-art is summarized and main contributions of this paper are highlighted.  
71 In section 2, parametric subspace-based modelling of 3D joint-trajectory is  
72 discussed. In Section 3, statistical tools developed on a Grassmann manifold  
73 are presented and a new supervised learning algorithm is introduced. In  
74 Section 4, the strength of the framework in term of accuracy and latency on  
75 several datasets are demonstrated. Finally, concluding remarks are presented  
76 in Section 5.

## 77 **2. Related works**

78 In this section two categories of related works are reviewed from two  
79 points of view: manifold-based approach and depth data representation.

80 We first review some related manifold based approaches for action analysis  
81 and recognition in 2D video. Then we focus on the most recent methods of  
82 action recognition from depth cameras.

### 83 *2.1. Manifold approaches in 2D videos*

84 Human action modelling from 2D video is a well studied problem in the  
85 literature. Recent surveys can be found in the work of Aggarwal et al. [12],  
86 Weinland et al. [13], and Poppe [4]. Beside classical methods performed  
87 in Euclidean space, a variety of techniques based on manifold analysis are  
88 proposed in recent years.

89 In the first category of manifold based approaches, each frame of action  
90 sequence (pose) is represented as an element of a manifold and the whole  
91 action is represented as a trajectory on this manifold. These approaches give  
92 solutions in the temporal domain to be invariant to speed and time using  
93 techniques like Dynamic Time Warping (DTW) to align action trajectories  
94 on the manifold. Also probabilistic grammatical models like Hidden Markov  
95 Model (HMM) are used to classify these actions presented as trajectories.  
96 Indeed, Veeraraghavan et al. [14] propose the use of human silhouettes ex-  
97 tracted from video images as a representation of the pose. Silhouettes are  
98 then characterized as points on the shape space manifold and modelled by  
99 ARMA models in order to compare sequences using a DTW algorithm. In  
100 another manifold shape space, Abdelkader et al. [15] represent each pose  
101 silhouette as a point on the shape space of closed curves and each gesture is  
102 represented as a trajectory on this space. To classify actions, two approaches  
103 are used: a template-based approach (DTW) and a graphical model approach  
104 (HMM). Other approaches use skeleton as a representation of each frame, as

105 works presented by Gong et al. [16]. They propose a spatio-Temporal Man-  
106 ifold (STM) model to analyze non-linear multivariate time series with latent  
107 spatial structure and apply it to recognize actions in the joint-trajectories  
108 space. Based on STM, they propose a Dynamic Manifold Warping (DMW)  
109 and a motion similarity metric to compare human action sequences both in  
110 2D space using a 2D tracker to extract joints from images and in 3D space  
111 using Motion capture data. Recently, Gong et al. [17] propose a Kernelized  
112 Temporal Cut (KTC) as an extension of their previous work [16]. They incor-  
113 porate Hilbert space embedding of distributions to handle the non-parametric  
114 and high dimensionality issues.

115 Some manifold approaches represent the entire action sequence as a point  
116 on an other special manifold. Indeed, Turaga et al. [18] involve a study of  
117 the geometric properties of the Grassmann and Stiefel manifolds, and give  
118 appropriate definitions of Riemannian metrics and geodesics for the purpose  
119 of video indexing and action recognition. Then, in order to perform the clas-  
120 sification as a probability density function, a mean and a standard-deviation  
121 are learnt for each class on class-specific tangent spaces. Turaga et al. [19]  
122 use the same approach to represent complex actions by a collection of sub-  
123 sequence. These sub-sequences correspond to a trajectory on a Grassmann  
124 manifold. Both DTW and HMM are used for action modelling and com-  
125 parison. Guo et al. [20] use covariance matrices of bags of low-dimensional  
126 feature vectors to model the video sequence. These feature vectors are ex-  
127 tracted from segments of silhouette tunnels of moving objects and coarsely  
128 capture their shapes.

129 Without any extraction of human descriptor as silhouette and neither an



130 explicit learning, Lui et al. [21] introduce the notion of tangent bundle to  
131 represent each action sequence on the Grassmann manifold. Videos are ex-  
132 pressed as a third-order data tensor of raw pixel from action images, which  
133 are then factorized on the Grassmann manifold. As each point on the mani-  
134 fold has an associated tangent space, tangent vectors are computed between  
135 elements on the manifold and obtained distances are used for action clas-  
136 sification in a nearest neighbour fashion. In the same way, Lui et al. [22]  
137 factorize raw pixel from images by high-order singular value decomposition  
138 in order to represent the actions on Stiefel and Grassmann manifolds. How-  
139 ever, in this work where raw pixels are directly factorized as manifold points,  
140 there is no dynamic modelling of the sequence. In addition, only distances  
141 obtained between all tangent vectors are used for action classification and  
142 there is no training process on data.

143 Kernels [23, 24] are also used in order to transform subspaces of a man-  
144 ifold onto a space where Euclidean metric can be applied. Shirazi et al.  
145 [23] embed Grassmann manifolds upon a Hilbert space to minimize cluster-  
146 ing distortions and then apply a locally discriminant analysis using a graph.  
147 Video action classification is then obtained by a Nearest-Neighbour classi-  
148 fier applied on Euclidean distances computed on the graph-embedded kernel.  
149 Similarly, Harandi et al. [24] propose to represent the spatio-temporal as-  
150 pect of the action by subspaces elements of a Grassmann manifold. Then,  
151 they embed this manifold into reproducing kernel of Hilbert spaces in order  
152 to tackle the problem of action classification on such manifolds. Gall et al.  
153 [25] use multi-view system coupling action recognition on 2D images with  
154 3D pose estimation, where the action-specific manifolds are acting as a link

155 between them.

156 All these approaches cited above are based on features extracted from  
157 2D video sequences as silhouettes or raw pixels from images. However, the  
158 recent emergence of low-cost depth sensors opens the possibility of revisiting  
159 the problem of activity modelling and learning using depth data-driven.

## 160 2.2. Depth data-driven approaches

161 Maps obtained by depth sensors are able to provide additional body shape  
162 information to differentiate actions that have similar 2D projections from a  
163 single view. It has therefore motivated recent research works, to investigate  
164 action recognition using the 3D information. Recent surveys [26, 27] are re-  
165 porting works on depth videos. First methods used for activity recognition  
166 from depth sequences have tendency to extrapolate techniques already de-  
167 veloped for 2D video sequences. These approaches use points in depth map  
168 sequences as a gray pixels in images to extract meaningful spatiotemporal  
169 descriptors. In Wanqing et al. [28], depth maps are projected onto the three  
170 orthogonal Cartesian planes ( $X - Y$ ,  $Z - X$ , and  $Z - Y$  planes) and the  
171 contours of the projections are sampled for each frame. The sampled points  
172 are used as *bag-of-points* to characterize a set of salient postures that corre-  
173 spond to the nodes of an *action graph* used to model explicitly the dynamics  
174 of the actions. Local feature extraction approaches like spatiotemporal inter-  
175 est points (STIP) are also employed for action recognition on depth videos.  
176 Bingbing et al.[29] use depth maps to extract STIP and encode Motion His-  
177 tory Image (MHI) in a framework combining color and depth information.  
178 Xia et al [30] propose a method to extract STIP a on depth videos (DSTIP).  
179 Then around these points of interest they build a depth cuboid similarity

180 feature as descriptor for each action. In the work proposed by Vieira et al.  
181 [31], each depth map sequence is represented as a 4D grid by dividing the  
182 space and time axes into multiple segments in order to extract SpatioTempo-  
183 ral Occupancy Pattern features (STOP). Also in Wang et al. [32], the action  
184 sequence is considered as a 4D shape but Random Occupancy Pattern (ROP)  
185 is used for features extraction. Yang et al.[33] employ Histograms of Oriented  
186 Gradients features (HOG) computed from Depth Motion Maps (DMM), as  
187 the representation of an action sequence. These histograms are then used as  
188 input to SVM classifier. Similarly, Oreifej et al. [34] compute a 4D histogram  
189 over depth, time, and spatial coordinates capturing the distribution of the  
190 surface normal orientation. This histogram is created using 4D projectors  
191 allowing quantification in 4D space.

192 The availability of 3D sensors has recently made possible to estimate 3D  
193 positions of body joints. Especially thanks to the work of Shotton et al.  
194 [6], where a real-time method is proposed to accurately predict 3D positions  
195 of body joints. Thanks to this work, skeleton based methods have become  
196 popular and many approaches in the literature propose to model the dynamic  
197 of the action using these features.

198 Xia et al. [35] compute histograms of the locations of 12 3D joints as a  
199 compact representation of postures and use them to construct posture visual  
200 words of actions. The temporal evolutions of those visual words are modeled  
201 by a discrete HMM. Yang et al. [36] extract three features, as pair-wise dif-  
202 ferences of joint positions, for each skeleton joint. Then, principal component  
203 analysis (PCA) is used to reduce redundancy and noise from feature, and it  
204 is also used to obtain a compact *Eigen Joints* representation for each frame.

205 Finally, a naïve-Bayes nearest-neighbour classifier is used for multi-class ac-  
206 tion classification. The popular Dynamic Time Warping (DTW) technique  
207 [37], well-known in speech recognition area, is also used for gesture and action  
208 recognition using depth data. The classical DTW algorithm was defined to  
209 match temporal distortions between two data trajectories, by finding an op-  
210 timal warping path between the two time series. The feature vector of time  
211 series is directly constructed from human body joint orientation extracted  
212 from depth camera or 3D Motion Capture sensors. Reyes et al. [38] per-  
213 form DTW on a feature vector defined by 15 joints on a 3D human skeleton  
214 obtained using PrimeSense NiTE. Similarly, Sempena et al. [39], by the 3D  
215 human skeleton model, use quaternions to form a 60-element feature vec-  
216 tor. The obtained warping path, by classical DTW algorithm, between two  
217 time series is mainly subjected to some constraints: (1) boundary constraint  
218 which enforces the first elements of the sequences as well as the last one  
219 to be aligned to each other (2) monotonicity constraint which requires that  
220 the points in the warping path are monotonically spaced in time in the two  
221 sequences. This technique is relatively sensitive to noise as it requires all  
222 elements of the sequences to be matched to a corresponding elements of the  
223 other sequence. It also has a drawback related to its computational complex-  
224 ity incurring in quadratic cost. However, many works have been proposed to  
225 bypass its drawbacks by means of probabilistic models [40] or incorporating  
226 manifold learning approach [17, 16].

227 Recent research has carried on more complex challenge of in-line recogni-  
228 tion systems for different applications, in which a trade-off between accuracy  
229 and latency can be highlighted. Ellis et al. [41] study this trade-off and

230 employed a Latency Aware Learning (LAL) method, reducing latency when  
231 recognizing actions. They train a logistic regression-based classifier, on 3D  
232 joint position sequences captured by kinect camera, to search a single canon-  
233 ical posture for recognition. Another work is presented by Barnachon et  
234 al. [42], where a histogram-based formulation is introduced for recognizing  
235 streams of poses. In this representation, classical histogram is extended to  
236 integral one to overcome the lack of temporal information in histograms.  
237 They also prove the possibility of recognizing actions even before they are  
238 completed using the integral histogram approach. Tests are made on both 3D  
239 MoCap from TUM kitchen dataset [43] and RGB-D data from MSR-Action  
240 dataset [28].

241 Some hybrid approaches combining both skeleton data features and depth  
242 information were recently introduced, trying to combine positive aspects of  
243 both approaches. Azary et al. [44] propose spatiotemporal descriptors as  
244 time-invariant action surfaces, combining image features extracted using ra-  
245 dial distance measures and 3D joint tracking. Wang et al. [45] compute  
246 local features on patches around joints for human body representation. The  
247 temporal structure of each joint in the sequence is represented through a tem-  
248 poral pattern representation called *Fourier Temporal Pyramid*. In Oreifej et  
249 al. [34], a spatiotemporal histogram (HON4D) computed over depth, time,  
250 and spatial coordinates is used to encode the distribution of the surface nor-  
251 mal orientation. Similarly to Wang et al. [45], HON4D histograms [34] are  
252 computed around joints to provide the input of an SVM classifier. Althloothi  
253 et al. [46] represent 3D shape features based on spherical harmonics repre-  
254 sentation and 3D motion features using kinematic structure from skeleton.

255 Both feature are then merged using multi kernel learning method.

256 It is important to note that, to date, few works have very recently pro-  
257 posed to use manifold analysis for 3D action recognition. Devanne et al. [47],  
258 propose a spatiotemporal motion representation to characterize the action as  
259 a trajectory which corresponds to a point on Riemannian manifold of open  
260 curves shape space. These motion trajectories are extracted from 3D joints,  
261 and the action recognition is performed by K-Nearest-Neighbor method ap-  
262 plied on geodesic distances obtained on open curve shape space. Azary et al.  
263 [48] use a Grassmannian representation as an interpretation of depth motion  
264 image (DMI) computed from depth pixel values. All DMI in the sequence  
265 are combined to create a motion depth surface representing the action as a  
266 spatiotemporal descriptor.

### 267 *2.3. Contributions and proposed approach*

268 On the one hand, approaches modelling actions as elements of manifolds  
269 [49, 50, 9] prove that it is an appropriate way to represent and compare  
270 videos. On the other hand, very few works deal with this task using depth  
271 images and it is still possible to improve learning step using these models.  
272 Besides, linear dynamic systems [51] show more and more promising results  
273 on the motion modelling since they exhibit the stationary properties in time,  
274 so they fit for action representation.

275 In this paper, we propose the use of geometric structure inherent in the  
276 Grassmann manifold for action analysis. We perform action recognition by  
277 introducing a manifold learning algorithm in conjunction with dynamic mod-  
278 elling process. In particular, after modelling motions as a linear dynamic sys-  
279 tems using ARMA models, we are interested in a representation of each point

280 on the manifold incorporating class separation properties. Our representa-  
281 tion takes benefit of statistics in the Grassmann manifold and action classes  
282 representations on tangent spaces. From spatiotemporal point of view, each  
283 action sequence is represented in our approach as linear dynamical system  
284 acquiring the time series of 3D joint-trajectory. From geometrical point of  
285 view, each action sequence is viewed as a point on the Grassmann manifold.  
286 In terms of machine learning, a discriminative representation is provided for  
287 each action thanks to a set of appropriate tangent vectors taking benefit  
288 of manifold proprieties. Finally, the efficiency of the proposed approach is  
289 demonstrated on three challenging action recognition datasets captured by  
290 depth cameras.

### 291 **3. Spatiotemporal modelling of action**

292 The human body can be represented as an articulated system composed  
293 of hierarchical joints that are connected with bones, forming a skeleton. The  
294 two best-known skeletons provided by the Microsoft Kinect sensor, are those  
295 obtained by official Microsoft SDK, which contains 20 joints, and PrimeSense  
296 NiTE which contains only 15 joints (see Figure 2). The various joint con-  
297 figurations throughout the motion sequence produce a time series of skeletal  
298 poses giving the skeleton movement. In our approach, an action is simply  
299 described as a collection of time series of 3D positions of the joints in the  
300 hierarchical configuration.

#### 301 *3.1. Linear dynamic model*

302 Let  $p_t^j$  denote the 3D position of a joint  $j$  at a given frame  $t$  i.e.,  $p^j =$   
303  $[x^j, y^j, z^j]_{j=1:J}$ , with  $J$  is the number of joints. The joint position time-series

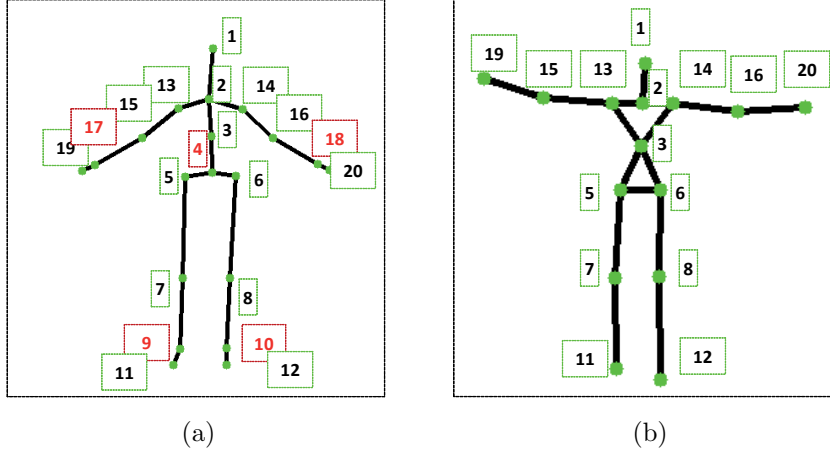


Figure 2: Skeleton joint locations captured by Microsoft Kinect sensor (a) using Microsoft SDK (b) using PrimeSense NiTE. Joint signification are: (1) head (2) shoulder center (3) spine (4) hip center (5/6) left/right hip (7/8) left/ ight knee (9/10) left/right ankle (11/12) left/right foot (13/14) left/right shoulder (15/16) left/right elbow (17/19) left/right wrist (18/20) left/right hand.

304 of joint  $j$  is  $p_t^j = \{x_t^j, y_t^j, z_t^j\}_{t=1:T}$ , with  $T$  the number of frames. A motion  
 305 sequence can then be seen as a matrix collecting all time-series from  $J$  joints,  
 306 i.e.,  $M = [p^1 p^2 \dots p^T]$ ,  $p \in \mathbb{R}^{3*J}$ .

307 At this level, we could consider using DTW algorithm [37] to find optimal  
 308 non-linear warping function to match these given time-series as proposed by  
 309 [38, 39, 16]. However, we opted for a system combining a linear dynamic  
 310 modelling with statistical analysis on a manifold, avoiding the boundary and  
 311 the monotonicity constraints presented by classical DTW algorithm. Such a  
 312 system is also less sensitive to noise due to the poor estimation of the joint  
 313 locations, in addition to its reduced computational complexity.

314 The dynamic and the continuity of movement imply that the action can  
 315 not be resumed as a simply set of skeletal poses because of the temporal



316 information contained in the sequence. Instead of directly using original  
 317 joint position time-series data, we believe that a linear dynamic system, like  
 318 that often used for dynamic texture modelling, is essential before manifold  
 319 analysis. Therefore, to capture both the spatial and the temporal dynamics  
 320 of a motion, linear dynamical system characterized by ARMA models are  
 321 applied to the 3D joint position time-series matrix  $M$ .

322 The dynamic captured by the ARMA [52, 53] model during an action  
 323 sequence  $M$  can be represented as:

$$\begin{aligned} p(t) &= Cz(t) + w(t), & w(t) &\sim N(0, R), \\ z(t+1) &= Az(t) + v(t), & v(t) &\sim N(0, Q) \end{aligned} \tag{1}$$

324 where  $z \in \mathbb{R}^d$  is a hidden state vector,  $A \in \mathbb{R}^{d \times d}$  is the transition matrix  
 325 and  $C \in \mathbb{R}^{3*J \times d}$  is the measurement matrix.  $w$  and  $v$  are noise components  
 326 modeled as normal with mean equal to zero and covariance matrix  $R \in$   
 327  $\mathbb{R}^{3*J \times 3*J}$  and  $Q \in \mathbb{R}^{d \times d}$  respectively. The goal is to learn parameters of the  
 328 model  $(A, C)$  given by these equations. Let  $U \Sigma V^T$  be the singular value  
 329 decomposition of the matrix  $M$ . Then, the estimated model parameters  $A$   
 330 and  $C$  are given by:  $\hat{C} = U$  and  $\hat{A} = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1}$ , where  
 331  $D_1 = [0 \ 0, I_{\tau-1} \ 0]$ ,  $D_2 = [I_{\tau-1} \ 0, 0 \ 0]$  and  $I_{\tau-1}$  is the identity matrix of  
 332 size  $\tau - 1$ .

333 Comparing two ARMA models can be done by simply comparing their  
 334 observability matrices. The expected observation sequence generated by an  
 335 ARMA model  $(A, C)$  lies in the column space of the extended observability  
 336 matrix given by  $\theta_\infty^T = [C^T, (CA)^T, (CA^2)^T, \dots]^T$ . This can be approximated  
 337 by the finite observability matrix  $\theta_m^T = [C^T, (CA)^T, (CA^2)^T, \dots, (CA^m)^T]^T$

338 [18]. The subspace spanned by columns of this finite observability matrix  
339 corresponds to a point on a Grassmann manifold.

### 340 3.2. Grassmann manifold interpretation

341 Grassmannian analysis provides a natural way to deal with the problem of  
342 sequence matching. Especially, this manifold allows to represent a sequence  
343 by a point on its space and offers tools to compare and to do statistics on  
344 this manifold. The classification problem of sets of motions represented by a  
345 collection of features can be transformed to point classification problem on  
346 the Grassmann manifold.

347 In this work we are interested in Grassmann manifolds which definition  
348 is as below.

349 *Definition:* The Grassmann manifold  $G_{n \times d}$  is a quotient space of orthogonal  
350 group  $O(n)$  and is defined as the set of  $d$ -dimensional linear subspaces of  $\mathbb{R}^n$ .  
351 Points on the Grassmann manifold are equivalent classes of  $n \times d$  orthogonal  
352 matrices, with  $d < n$ , where two matrices are equivalent if their columns span  
353 the same  $d$ -dimensional subspace.

354 Let  $\mu$  denotes an element on  $G_{n \times d}$ , the tangent space to this element  $T_\mu$  on  
355  $G_{n,d}$  is the tangent plane to the surface of the manifold at  $\mu$ . It is possible  
356 to map a point  $U$ , of the Grassmann manifold, to a vector in the tangent  
357 space  $T_\mu$  using the logarithm map as defined by Turaga et al. [18]. An other  
358 important tool in statistics is the exponential map  $Exp_\mu : T_\mu(G_{n,d}) \rightarrow G_{n,d}$ ,  
359 which allows to move on the manifold.

360 Two points  $U_1$  and  $U_2$  on  $G_{n,d}$  are equivalent if one can be mapped into  
361 the other one by  $d \times d$  orthogonal matrix [54]. In other words,  $U_1$  and  $U_2$  are  
362 equivalent if the  $d$  columns of  $U_1$  are rotations of  $U_2$ . The minimum length

363 curve connecting these two points is the geodesic between them computed  
364 as:

$$d_{geod}(U_1, U_2) = \| [\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_d] \|_2 \quad (2)$$

365 where  $\theta_i$  is the principal angle vector which can be computed through the  
366 SVD of  $U_1^T U_2$ .

#### 367 4. Learning process on the manifold

368 Let  $\{U_1, \dots, U_N\}$  be  $N$  actions represented by points on the Grassmann  
369 manifold. A common learning approach on manifolds is based on the use  
370 of only one-tangent space, which usually can be obtained as the tangent  
371 space to the mean ( $\mu$ ) of the entire data points  $\{U_i\}_{i=1:N}$  without regard  
372 to class labels. All data points on the manifold are then projected on this  
373 tangent space to provide the input of a classifier. This assumption provide an  
374 accommodated solution to use a classical supervised learning on the manifold.  
375 However, this flattening of the manifold through tangent space is not efficient  
376 since the tangent space on the global mean can be far from other points.

377 A more appropriate way is to consider separate tangent spaces for each  
378 class at the class-mean. The classification is then performed in these indi-  
379 vidual tangent spaces as in [18].

380 Some other approaches explore the idea of tangent bundle as in Lui et  
381 al. [21, 22], in which all tangent planes of all data points on the manifold  
382 are considered. Tangent vectors are then computed between all points on  
383 a Grassmann manifold and action classification is performed thanks to ob-  
384 tained distances.

385 We believe that using several tangent spaces, obtained for each class of

386 the training data points, is more intuitive. However, the question here is how  
387 to learn a classifier in this case?

388 In the rest of the section, we present a statistical computation of the mean  
389 in the Grassmann manifold [55]. Then, we propose two learning methods on  
390 this manifold taking benefit from tangent space class specific and tangent  
391 bundle [21]: Truncated Wrapped Gaussian (TWG) [56] and Local Tangent  
392 Bundle SVM (LBTSVM).

#### 393 *4.1. Mean computation on the Grassmann manifold*

394 The Karcher mean [55] enables computation of a mean representative for  
395 a cluster of points on the manifold. This mean should belong to the same  
396 space as the given points. In our case, we need Karcher mean to compute  
397 averages on the Grassman manifold and more precisely means of each action  
398 class which represents the action at best. The algorithm exploits *log* and *exp*  
399 maps in a predictor/corrector loop until convergence to an expected point.

400 The computation of a mean can be used to perform an action classification  
401 solution. This can be done by a simple comparison of an unknown action,  
402 represented as a point on the manifold, to all class-means and assigning it to  
403 the nearest one using the distance presented in Equation 2.

#### 404 *4.2. Truncated Wrapped Gaussian*

405 In addition to the mean  $\mu$  computed by Karcher mean on  $\{U_i\}_{i=1:N}$ , we  
406 look for the standard deviation value  $\sigma$  between all actions in each class of  
407 training data. The  $\sigma$  must be computed on  $\{V_i\}_{i=1:N}$  where  $V = \exp_{\mu}^{-1}(U_i)$   
408 are the projections of actions from the Grassmann manifold into the tangent

409 space defined on the mean  $\mu$ . The key idea here is to use the fact that the  
 410 tangent space  $T_\mu(G_{n,d})$  is a vector space.

411 Thus, we can estimate the parameters of a probability density function  
 412 such as a Gaussian and then use the exponential map to wrap these param-  
 413 eters back onto the manifold using exponential map operator [18]. However,  
 414 the exponential map is not a bijection for the Grassmann manifold. In fact, a  
 415 line on tangent space, with infinite length, can be warped around the man-  
 416 ifold many times. Thus, some points of this line are going to have more than  
 417 one image on  $G_{n,d}$ . It becomes a bijection only if the domain is restricted.  
 418 Therefore, we can restrict the tangent space by a truncation beyond a radius  
 419 of  $\pi$  in  $T_\mu(G_{n,d})$ . By truncation, the normalization constant changes for mul-  
 420 tivariate density in  $T_\mu(G_{n,d})$ . In fact, it gets scaled down depending on how  
 421 much of the probability mass is left out of the truncation region.

422 Let  $f(x)$  denotes the probability density function (pdf) defined on  $T_\mu(G_{n,d})$   
 423 by :

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

424 After truncation, an approximation of  $f$  gives:

$$\hat{f}(x) = \frac{f(x) \times \mathbf{1}_{|x| < \pi}}{z} \quad (4)$$

425 where  $z$  is the normalization factor :

$$z = \int_{-\pi}^{\pi} f(x) \times \mathbf{1}_{|x| < \pi} dx \quad (5)$$

426 Using Monte Carlo estimation, it can proved that the estimation of  $z$  is given

427 by:

$$\hat{z} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{|x_i| < \pi} \quad (6)$$

428 In practice, we employ wrapped Gaussians in each class-specific tangent  
429 space. Separate tangent space is considered for each class at its mean com-  
430 puted by Karcher mean algorithm. Predicted class of an observation point  
431 is estimated in these individual tangent spaces. In the training step, the  
432 mean, standard deviation and normalization factor in each class of actions  
433 are computed. The predicted label of unknown class action is estimated as  
434 a function of probability density in class-specific tangent spaces.

#### 435 *4.3. Local Tangent Bundle*

436 We intent here to generalize a learning algorithm to work with data points  
437 which are geometrically lying to a Grassmann manifold. Using multiple class-  
438 specific tangent spaces is decidedly more relevant than single one. However,  
439 restrict the learning to only the mean and the standard-deviation in each tan-  
440 gent space, as in TGW method, is probably insufficient to classify complex  
441 actions with small inter-class variation. Our idea is to build a supervised clas-  
442 sifier on the manifold but without limiting the learning process to distances  
443 computed on the tangent spaces as in [22].

444 We consider such data points to be embedded in higher dimensional rep-  
445 resentation providing a natural and implicit separation of directions. We  
446 use the notion of tangent bundle on the manifold to formulate our learning  
447 algorithm.

448 The tangent bundle of a manifold is defined in the literature as the mani-  
449 fold along with the set of tangent planes taken at all points on it. Each such

450 a tangent plane can be equipped with a local Euclidean coordinate system.  
451 In our approach, we consider several "local" bundles, each one represents the  
452 tangent planes taken at all points belonging to a class from training dataset  
453 and expressed as class-specific local bundle.

454 We generate Control Tangents (CT) on the manifold, which represent  
455 all class-specific local bundles of data points. Each CT can be seen as the  
456 tangent space of the Karcher mean of all points belonging to the same class  
457 of points from only training data. Karcher mean algorithm can be employed  
458 here for mean computation.

459 We introduce an upswing of the manifold learning so-called Local Tangent  
460 Bundle (LTB), in which proximities are required between each point on the  
461 manifold and all CTs. The LTB can be viewed as a parameterization of  
462 a point on the manifold which incorporates implicitly release properties in  
463 relation to all class clusters, by mapping this point to all CTs using logarithm  
464 map.

465 The LTBs can provide the input of a classifier, like the linear SVM clas-  
466 sifier as in our case. In doing so, the learning model of the classifier is con-  
467 structed using LTBs instead of classifying as function of the local distances  
468 (mean and standard-deviation) of the point from LTBs as in TWG method.

469 We finally notice that training a linear SVM classifier on our represen-  
470 tation of points provided by LTB is more appropriate than the use of SVM  
471 with classical Kernel, like rbf, on original points on the manifold.

472 In experiments, we compare our learning approach LTBSVM to the clas-  
473 sical one denoted as One-tangent SVM (TSVM), in which the mean is com-  
474 puted on the entire training dataset regardless to class labels. Then, all

475 points on the manifold are projected on this later to provide the inputs of a  
 476 linear SVM.

477 A graphical illustration of the manifold learning by TWG and LTB can  
 478 be shown in Figure 3.

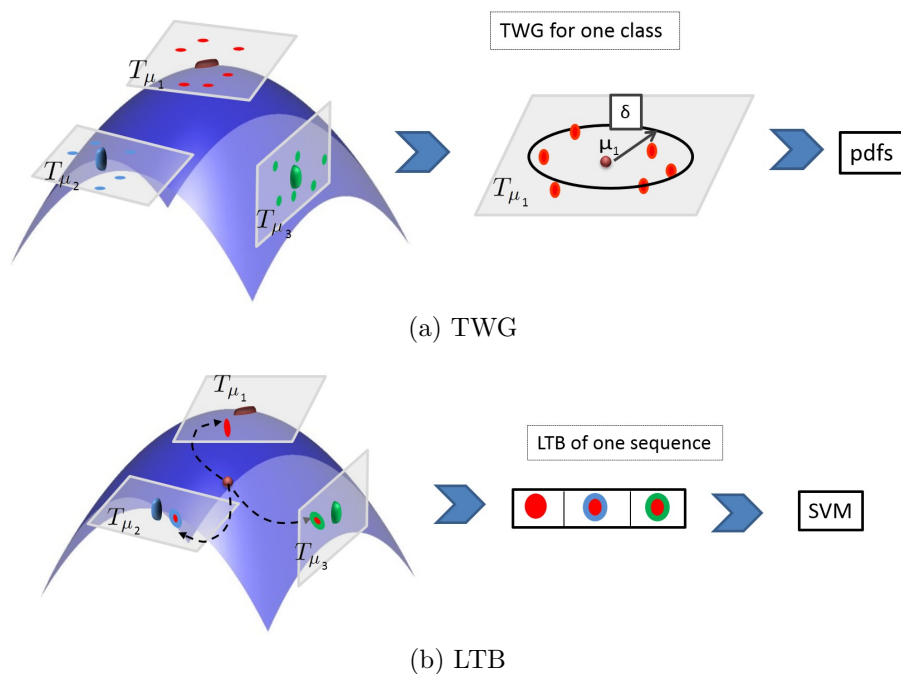


Figure 3: Conceptual TWG and LTB learning methods on the Grassmann manifold. (a) Actions belonging to the same class, illustrated with same color, are projected to the tangent space presented with their mean and then Gaussian function is computed on each tangent space, (b) An action is projected on all CTs, and thus construct a new observation is represented by its LTB.

## 479 5. Experimental results

480 This section summarizes our empirical results and provides an analysis of  
 481 the performances of our proposed approach on several datasets compared to  
 482 the state-of-the-art approaches.



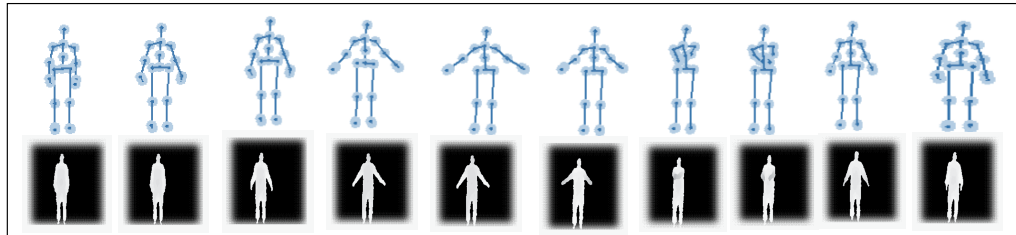
483 *5.1. Data and features*

484 We extensively experimented our proposed approach on three public 3D  
 485 action datasets containing various challenges, including MSR-action 3D [28],  
 486 UT-kinect [35] and UCF-kinect [41]. All details about these datasets: differ-  
 487 ent types and number of motions, number of subjects executing these motions  
 488 and the experimental protocol used for evaluation are summarized in Table  
 489 1. Examples of actions from these datasets are shown in Figure 4.

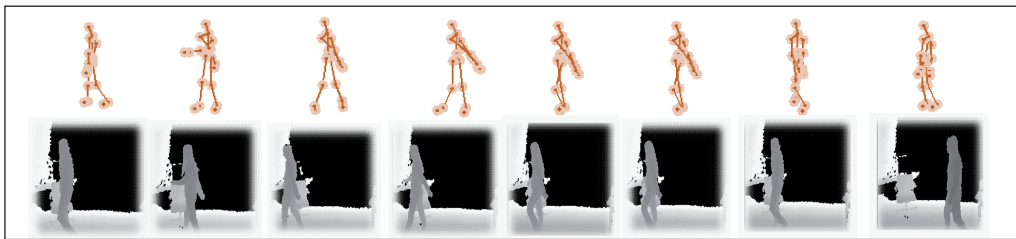
Dataset	Motions	Total number of ac- tions	Experimental protocol
MSR-action 3D [28]	RGB + depth (320*240) + 20 joints: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw X, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up and throw	10 subjects   20 ac- tions   3 try $\Rightarrow$ To- tal of 520 actions	50% Learn- ing / 50% Testing
UT-kinect [35]	RGB + depth (320*240) + 20 joints: walk, sit down, stand up, pick up, carry, throw, push, pull, wave and clap hands	10 subject   10 ac- tions   2 try $\Rightarrow$ To- tal of 200 actions	leave-one- out cross- validation
UCF-kinect [41]	15 joints: balance, climb up, climb ladder, duck, hop, vault, leap, run, kick, punch, twist left, twist right , step forward, step back, step left, step right	16 subjects   16 ac- tions   5 try $\Rightarrow$ To- tal of 1280 actions	70% Learn- ing / 30% Testing

Table 1: Overview of the datasets used in the experiments.

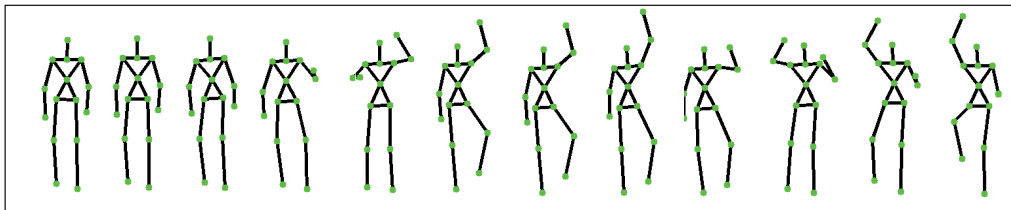
490 In all these datasets, a normalization step is performed in order to make  
 491 the skeletons scale-invariant. For each frame, the hip center joint is first  
 492 placed at the origin of the coordinate system. Then, a skeleton template is  
 493 taken as reference and all the other skeletons are normalized such that their



(a) MSR-action 3D



(b) UT-Kinect



(c) UCF-kinect

Figure 4: Examples of human actions from datasets used in our experiments: (a) 'hand clap' from MSR-action 3D , (b) 'walk' from UT kinect and (c) 'climb ladder' from UCF-kinect.

494 body part lengths are equal to the corresponding lengths of the reference  
495 skeleton. Each 3D joint sequence is represented as time series matrix of size  
496  $F \times T$  with  $T$  the number of frames in the sequence and  $F$  the number  
497 of features per frame. The number of features depends on the number of  
498 estimated joints (60 values for Microsoft SDK skeleton and 45 for PrimeSense  
499 NiTE skeleton). The dynamic of the activity is then captured using an  
500 ARMA model. In this process, a dimensionality reduction is needed and best  
501 subspace dimension "d" have been chosen using a 5-fold cross-validation on  
502 the training dataset. The parameter giving the best accuracy on the training  
503 set is kept for all experiments.

504 Each action is an element of the Grassmann manifold  $G_{n \times d}$  with  $n = m \times J$   
505 where  $J$  represents the number of joints and  $d$  is the subspace dimension  
506 learnt on the training data. We set  $m = d$ , while  $m$  represents the truncation  
507 parameter of observation.

508 In our LTBSVM approach, we train a linear SVM on our LTB represen-  
509 tations of points on the Grassmann manifold. We use a multi-class SVM  
510 classifier from LibSVM library [57], where the penalty parameter  $C$  is tuned  
511 using a 5-fold cross-validation on the training dataset.

512 We evaluate the performance of our approach for action recognition and  
513 explore the latency on recognition by evaluating the trade-off between accu-  
514 racy and latency over varying number of actions. To allow a better evalua-  
515 tion of our approach, we conducted experiments respecting those made in the  
516 state-of-the-art approaches. We note here that other interesting datasets are  
517 available, like TUM kitchen dataset [43] which presents challenging short and  
518 complex actions. In our experiments we concentrated on three other datasets

519 from depth sensors (such as kinect), chosen according to the challenges they  
520 contain, as occlusion, change of view and possibility to compare the latency.  
521 Details of the experiments are presented in the following sections.

## 522 *5.2. MSR-Action 3D dataset*

523 MSR-Action 3D [28] is a public dataset of 3D action captured by a depth  
524 camera. It consists of a set of temporally segmented actions where subjects  
525 are facing the camera and they are advised to use their right arm or leg if  
526 an action is performed by a single limb. The background is pre-processed  
527 clearing discontinuities and there is no interaction with objects in performed  
528 actions. Despite of all of these facilities, it is also a challenging dataset  
529 since many activities appear very similar due to small inter-class variation.  
530 Several works have already been conducted on this dataset. Table 2 shows  
531 the accuracy of our approach compared to the state-of-the-art methods. We  
532 followed the same experimental setup as in Oreifej et al. [34] and Jiang et  
533 al. [45], where first five actors are used for training and the rest for testing.

534 Our results obtained in this table correspond to four learning methods:  
535 simple Karcher Mean (KM), one Tangent SVM (TSVM), Truncated Wrapped  
536 Gaussian (TWG) and Local Tangent Bundle SVM (LTBSVM). Our approach  
537 using LTBSVM achieves an accuracy of 91.21%, exceeding the best method  
538 from the state-of-the-art proposed by Oreifej et al. [34]. We note that our  
539 approach is based on only skeletal joint coordinates as motion features, com-  
540 pared to other approaches, such as Oreifej et al. [34] and Wang et al. [32]  
541 which use the depth map or depth information around joint locations.

542 To evaluate the effect of the changing of the subspace dimensions, we  
543 conduct several tests on MSR-Action 3D dataset with different dimensions

Method	accuracy %
Histograms of 3D Joints [58]	78.97
Eigen Joints [36]	82.33
DMM-HOG [33]	85.52
HON4D [34]	85.80
Random Occupancy patterns [32]	86.50
Actionlet Ensemble [45]	88.20
HOH4D + $D_{disc}$ [34]	88.89
TSVM on one tangent space	<b>74.32</b>
KM	<b>77.02</b>
TWG	<b>84.45</b>
LTBSVM	<b>91.21</b>

Table 2: Recognition accuracy (in %) for the MSR-Action 3D dataset using our approach compared to the previous approaches.

544 of subspaces. Figure 5 shows the variation of recognition performances with  
545 the change of the subspace dimension. We remark that until dimension 12,  
546 the recognition rate generally increase with the increase of the size of the  
547 subspaces dimensions. This is expected, since a small dimension causes a  
548 lack of information but also a big dimension of the subspace keeps noise and  
549 brings confusion between inter-classes. We also compare in this figure, our  
550 new introduced learning algorithm LBTSVM to TWG and KM.

551 To better understand the behavior of our approach according to the action  
552 type, the confusion matrix is illustrated in Figure 6. For most actions, about  
553 11 classes of actions, video sequences are 100% correctly classified.

554 The classification error occurs if two actions are very similar, such as  
555 'horizontal arm wave' and 'high arm wave'. Besides, one of most problematic  
556 action to classify is 'hammer' action which is frequently confused with 'draw  
557 X'. The particularity of these two actions is that they start in the same  
558 way but one finishes before the other. If we show only the first part of  
559 'draw X' action and the whole sequence of 'hammer' action we can see that

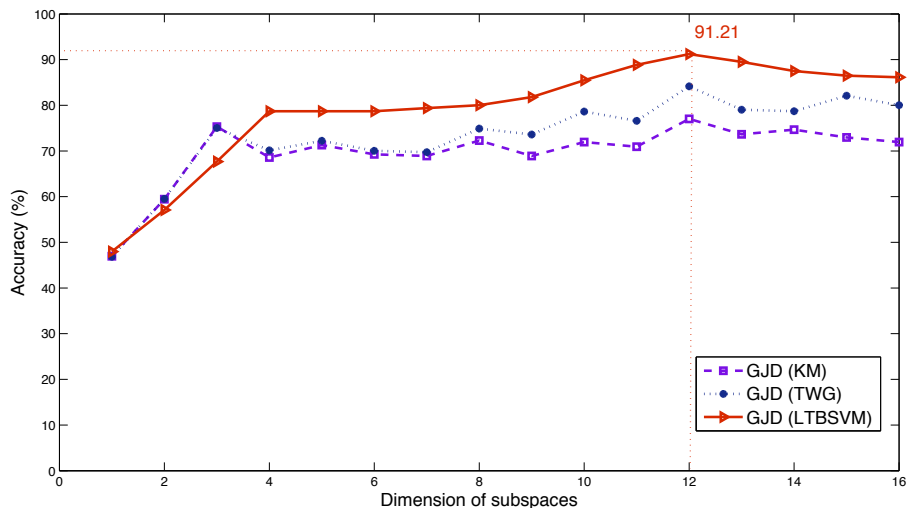


Figure 5: Recognition rate variation with learning approach and subspace dimension.

560 they are very similar. The same for 'hand catch' action which is confused  
 561 with 'draw circle'. It is important to note that 'hammer' action is completely  
 562 misclassified with the approach presented by Oreifej et al. [34] which presents  
 563 the second better recognition rate after our approach.

564 While the focus of this paper is mainly on action recognition and latency  
 565 reduction, some applications need to perform training step with a reduced  
 566 amount of data. To study the effect of the amount of training dataset, we  
 567 measured how the accuracy changed as we iteratively reduced the number of  
 568 actions per class in the training dataset. Table 3 shows obtained accuracy  
 569 results with different size of training dataset.

570 These results show that, in contrast to approaches that use HMM which  
 571 require a large number of training data, our approach reveals robustness and  
 572 efficiency. This robustness is due to the fact that the Control Tangents, which

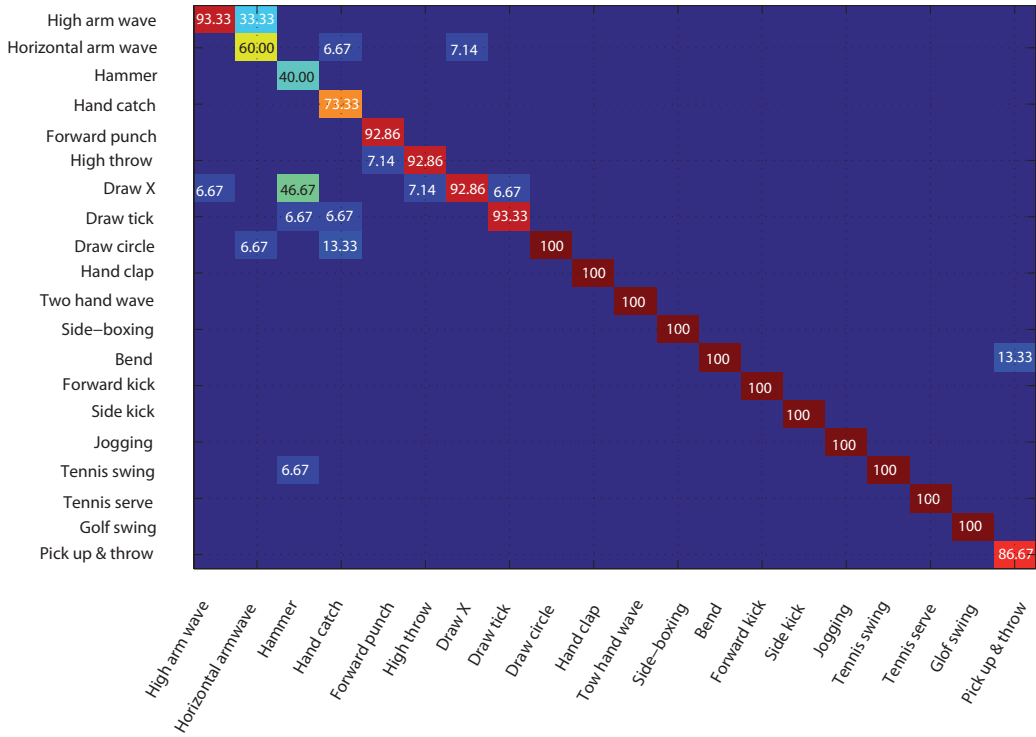


Figure 6: The confusion matrix for the proposed approach on MSR-Action 3D dataset. It is recommended to view the Figure on the screen.

573 play an important role in learning process, can be computed efficiently using  
 574 small number of action points per class on the manifold.

### 575 5.3. UT-Kinect dataset

576 Sequences of this dataset are taken using one depth camera (kinect) in  
 577 indoor settings and their length vary from 5 to 120 frames. We use this  
 578 dataset because it contains several challenges:

- 579 • View change, where actions are taken from different views: right view,  
 580 frontal view or back view.

Actions per class	Training dataset %	Accuracy %
5	37.17	73.36
6	44.23	77.64
7	51.13	83.10
8	58.36	84.79
9	65.54	88.51
10	72.49	89.18
11	79.95	87.83
12	86.24	88.85
13	91.07	90.20
14	95.91	90.54
15	100	91.21

Table 3: Recognition accuracy, obtained by our approach using LTBSVM on MSR-Action 3D dataset, with different size of training dataset.

- 581     • Significant variation in the realization of the same action: same action  
582       is done with one hand or two hands can be used to describe the 'pick  
583       up' action.
  
- 584     • Variation in duration of actions: the mean and standard-deviation are  
585       respectively for the whole actions 31.1 and 11.61 frames at 30 fps.

586 To compare our results with state-of-the-art approaches, we follow experi-  
587 ment protocol proposed by Xia et al. [35]. The protocol is leave-one-out  
588 cross-validation. In Table 4, we show comparison between the recognition  
589 accuracy produced by our approach and the approach presented by Xia et  
590 al. [35].

591 This table shows the accuracy of the five least-recognized actions in UT-  
592 kinect dataset and the five best-recognized actions. Our system performs  
593 the worst when the action represents an interaction with an object: 'throw',  
594 'push', 'sit down' and 'pick up'. However, for the best five recognized actions,  
595 our approach improves the recognition rate reaching 100%. These actions



Action	Acc % Xia et al. [35]	Acc % LTBSVM
Walk	96.5	<b>100</b>
Stand up	91.5	<b>100</b>
Pick up	97.5	<b>100</b>
Carry	97.5	<b>100</b>
Wave	100	<b>100</b>
Throw	59	60
Push	81.5	65
Sit down	91.5	80
Pull	92.5	85
Clap hands	100	95
Overall	90.92	88.5

Table 4: Recognition accuracy (per action) for the UT-kinect dataset obtained by our approach using LTBSVM compared to Xia et al. [35].

596 contain variations in view point and realization of the same action. This  
597 means that our approach is view-invariant and it is robust to change in action  
598 types thanks to the used learning approach. The overall accuracy of Xia et al.  
599 [35] is better than our recognition rate. However on MSR Action3D database,  
600 the recognition rate obtained by this approach gives only 78.97%. This can  
601 be explained by the fact that this approach requires a large training dataset.  
602 Especially for complex actions which affect adversely the HMM classification  
603 in case of small samples of training.

#### 604 5.4. UCF-kinect dataset

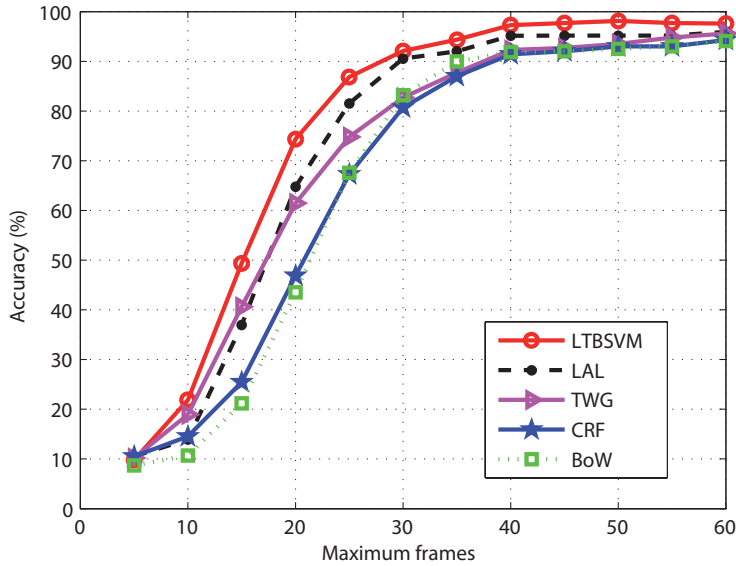
605 In this experiment, our approach is evaluated in terms of latency, i.e.  
606 the ability for a rapid (low-latency) action recognition. The goal here is to  
607 automatically determine when a sufficient number of frames are observed to  
608 permit a reliable recognition of the occurring action. For many applications,  
609 a real challenge is to define a good compromise between "making forced de-  
610 cision" on partial available frames (but potentially unreliable) and "waiting"

611 for the entire video sequence.

612 To evaluate the performance of our approach in reducing latency, we con-  
613 ducted our experiments on UCF-kinect dataset [41]. The skeletal joint loca-  
614 tions (15 joints) over sequences of this dataset are estimated using Microsoft  
615 Kinect sensor and the PrimeSense NiTE. The same experimental setup as  
616 in Ellis et al. [41] is followed. For a total of 1280 action samples contained  
617 in this dataset, a 70% and 30% split is used for respectively training and  
618 testing datasets. From the original dataset, new subsequences were created  
619 by varying a parameter corresponding to the  $K$  first frames. Each new sub-  
620 sequence was created by selecting only the first  $K$  frames from the video. For  
621 videos shorter than  $K$  frames, the entire video is used. We compare the re-  
622 sult obtained by our approach to those obtained by Latency Aware Learning  
623 (LAL) method proposed by Ellis et al. [41] and other baseline algorithms:  
624 Bag-of-Words (BoW) and Linear Chain Conditional Random Field (CRF),  
625 also reported by Ellis et al. [41].

626 As shown in Figure 7, our approach using LTBSVM clearly achieves im-  
627 proved latency performance compared to all other baseline approaches. Anal-  
628 ysis of these curves shows that, accuracy rates for all other approaches are  
629 close when using small number of frames (less than 10) or a large number of  
630 frames (more than 40). However, the difference increases significantly in the  
631 middle range. The table joint to Figure 7 shows numerical results at several  
632 points along the curves in the figure. Thus, given only 20 frames of input,  
633 our system achieves 74.37%, while BOW, CRF recognition rate below 50%  
634 and LAL achieves 61.45%.

635 It is also interesting to notice the improvement of accuracy of 92.08%



Approach/frames	10	15	20	25	30	40	60
LTBSVM	<b>21.87</b>	<b>49.37</b>	<b>74.37</b>	<b>86.87</b>	<b>92.08</b>	<b>97.29</b>	<b>97.91</b>
TWG	18.95	40.62	61.45	74.79	82.7	92.29	95.62
LAL [41]	13.91	36.95	64.77	81.56	90.55	95.16	95.94
CRF [41]	14.53	25.46	46.88	67.27	80.70	91.41	94.06
BOW [41]	10.7	21.17	43.52	67.58	83.20	91.88	94.06

Figure 7: Accuracy vs. state-of-the-art approaches over videos truncated at varying maximum lengths. Each point of this curve shows the accuracy achieved by the classifier given only the number of frames shown in the x-axis.

636 obtained by LTBSVM compared to 82.7% obtained by TWG, with maximum  
 637 frame number equal to 30. For a large number of frames, all of the methods  
 638 perform globally a good accuracy with an improvement of the ours (97.91%  
 639 comparing to 95.94% obtained by LAL proposed in Ellis et al. [41]). These  
 640 results show that our approach can recognize actions at the desired accuracy  
 641 with reducing latency.

642 Finally, the detail of recognition rates, when using the totality of frames  
 643 in the sequence, are shown through the confusion matrix in Figure 8. Unlike  
 644 what gives LAL, we can observe that the 'twist left', 'twist right' actions are

645 not confused with each others. All classes of actions are classified with a rate  
 646 more than 93.33% which gives a lot of confidence to our proposed learning  
 647 approach.

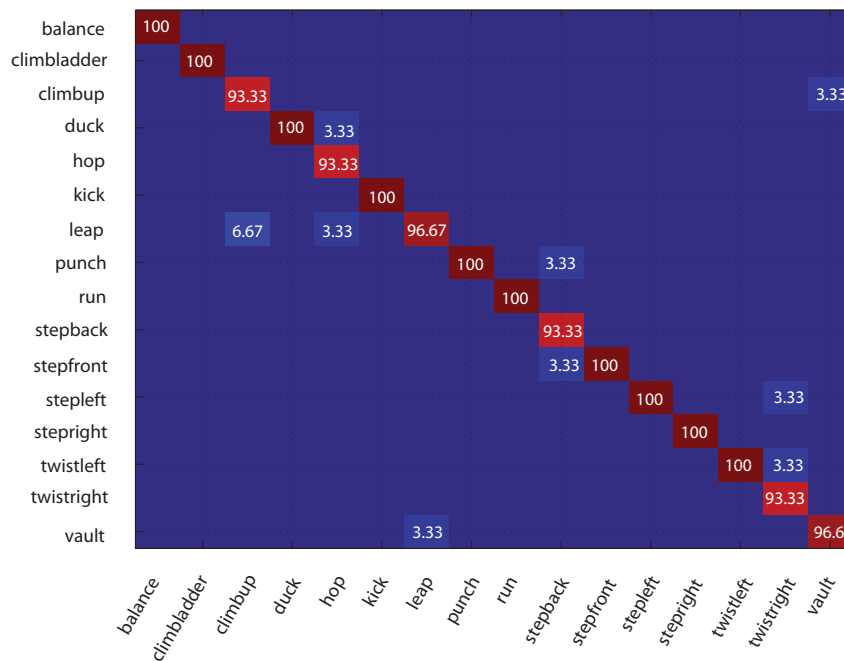


Figure 8: The confusion matrix for the proposed method on UCF-kinect dataset. Overall accuracy achieved 97.91%. It is recommended to view the figure on the screen.

## 648 5.5. Discussion

649 *Manifold representation and learning.* Data representation is one of the most  
 650 important factors in the recognition approach, on which we must take a lot  
 651 of consideration. Our data representation, like many state-of-the-art man-  
 652 ifold techniques [19, 14, 21], consider the geometric space and incorporates  
 653 the intrinsic nature of the data. In our framework, which is 3D joint-based,  
 654 both geometric appearance and dynamic of human body are captured simul-

655 taneously. Furthermore, unlike the manifold approaches using silhouettes  
656 [14, 15, 18], or directly raw pixels [22, 19], our approach use informative  
657 geometric features, which capture useful knowledge to understand the in-  
658 trinsic motion structure. Thanks to recent release of depth sensor, these  
659 features are extracted and tracked along the action sequence, while classical  
660 pixel-based manifold approaches relying on a good action localization, or on  
661 tedious feature extraction from 2D videos like silhouettes.

662 In terms of learning method, we generalized a learning algorithm to work  
663 with data points which are geometrically lying to a Grassmann manifold.

664 Other approaches are tested in the learning process on the manifold: one  
665 tangent space (TSVM) and class-specific tangent spaces (TWG). In the first  
666 one, recognition rate is low. In fact, the computation of the mean of all  
667 actions from all classes can be inaccurate. Besides, projections on this plane  
668 can lead to big deformations. A better solution is to operate on each class by  
669 computing its proper tangent space, as in TWG [56] which improve TSVM  
670 results (see Table 2). In our approach (LTBSVM), both Control Tangent  
671 and statistics on the manifold are used. The purpose was to formulate our  
672 learning algorithm using a discriminative parametrization which incorporate  
673 class separation properties. The particularity of our learning model is the  
674 incorporation of proximities relative to all Control Tangent spaces represent-  
675 ing class clusters, instead of classifying using a function of local distances.  
676 The results in Table 2 demonstrate that the proposed algorithm is more effi-  
677 cient in action recognition scenario when inter-variation classes is present as  
678 a challenge.

679 Furthermore, the analysis of the impact of reducing the number of actions

680 in the training set on the accuracy of the classifier show robustness. Even  
681 with a small number of actions in the training data recognition rates remain  
682 good as demonstrated in Table 3. However it is a limitation especially for  
683 approaches using an HMM learning because they require a large number of  
684 training dataset. Such as Xia et al. approach [35], which gives only 78.97%  
685 of recognition rate while performing cross subject test on MSR dataset.

686 *Latency and Time computation.* The evaluations in terms of latency have  
687 clearly revealed the efficiency of our approach for a rapid recognition. It  
688 is possible to recognize actions up to 95% using only 40 frames which is  
689 a good performance comparing to state-of-the-art approaches presented in  
690 [41]. Thus, our approach can be used for interactive systems. Particularly,  
691 in entertainment applications to resolve the problem of lag and improve some  
692 motion-based games.

693 Since the proposed approach is based on only skeletal joint coordinates,  
694 it is simple to calculate and it needs only a small computation time. In fact,  
695 with our current implementation written in C++, the whole recognition time  
696 takes 0.26 sec to recognize a sequence of 60 frames. The joint extraction and  
697 normalisation take 0.0001 sec, the Grassmann and the LTB representation  
698 take 0.0108 sec and the prediction on SVM takes 0.251 sec. These computa-  
699 tion time are reported on UCF dataset, with Grassmann manifold dimension  
700  $n = 540$  and  $d = 12$ . We also reported the computation time needed to  
701 recognize actions while incorporating latency on UCF dataset. Figure 9 il-  
702 lustrates inline time recognition with time progression, after only 40 frames  
703 the recognition is given at the 0.94 sec within 97.29% of correctness rate.  
704 After 60 frames, in 1.3 sec the algorithm recognize correctly the action with

705 97.91%. All the computation time experiments are lunched on a PC having  
 706 Intel Core i5-3350P (3.1 GHz) CPU, 4GB RAM and a PrimeSense camera  
 707 for skeleton extraction giving about 60 skeleton/sec.

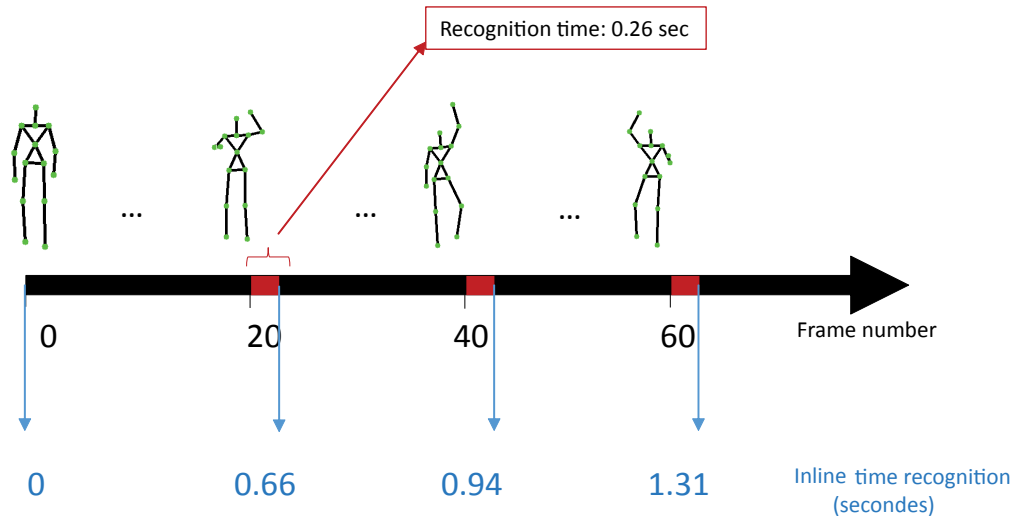


Figure 9: The computation time to perform 20 frames actions sequences is 0.26 sec by using our approach. The computation time is given for each actions frames sequences (e.g. 0.94 sec for 40 frames).

708 *Limitations.* Our proposed approach is a 3D joint-based framework derives  
 709 a human action recognition from skeletal joint sequences. In the case of  
 710 presence of object interaction in human actions, our approach do not provides  
 711 any relevant information about objects and thus, action with and without  
 712 objects are confused. This limitation can be leveraged in future by the use  
 713 of additional features, which can be extracted from depth or color images  
 714 associated to 3D joint locations.

## 715 **6. Conclusion**

716 In this paper, an effective framework for modelling and recognizing hu-  
717 man motion in the 3D skeletal joint space is proposed. In this framework,  
718 sequence features are modeled temporally as subspaces lying to a Grassman-  
719 nian manifold. A new learning algorithm on this manifold is then introduced.  
720 It embeds each action, presented as a point on the manifold, in higher dimen-  
721 sional representation providing natural separation directions. We formulated  
722 our learning algorithm using the notion of local tangent bundles on class clus-  
723 ters on the Grassmann manifold. The empirical results and the analysis of  
724 the performance of our proposed approach show promising results with high  
725 accuracies superior to 88% on three different datasets. The evaluation of  
726 our approach in terms of accuracy/latency reveals an important ability for  
727 a low-latency action recognition system. Obtained results show that with  
728 minimum number of frames, it provides the highest recognition rate.

729 We would encourage future works to extend our approach to investigate  
730 more challenging problems like human behaviour recognition. Finally, we  
731 plan to use additional features from depth or color images associated to 3D  
732 joint locations to solve the problem of human-object interaction.

## 733 **References**

- 734 [1] S. Fothergill, H. Mentis, P. Kohli, S. Nowozin, Instructing people for  
735 training gestural interactive systems, in: CHI Conference on Human  
736 Factors in Computing Systems, New York, NY, USA, 2012, pp. 1737–  
737 1746.



- 738 [2] W. Lao, J. Han, P. de With, Automatic video-based human motion  
739 analyzer for consumer surveillance system, in: IEEE Transactions on  
740 Consumer Electronics, Vol. 55, 2009, pp. 591–598.
- 741 [3] A. Jalal, M. Uddin, T. S. Kim, Depth video-based human activity recog-  
742 nition system using translation and scaling invariant features for life log-  
743 ging at smart home, in: IEEE Transactions on Consumer Electronics,  
744 Vol. 58, 2012, pp. 863–871.
- 745 [4] R. Poppe, A survey on vision-based human action recognition, in: Image  
746 and Vision Computing, Vol. 28, 2010, pp. 976–990.
- 747 [5] P. Turaga, R. Chellappa, V. S. Subrahmanian, O. Udrea, Machine recog-  
748 nition of human activities: A survey, in: IEEE Transactions on Circuits  
749 and Systems for Video Technology, Vol. 18, Piscataway, NJ, USA, 2008,  
750 pp. 1473–1488.
- 751 [6] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore,  
752 A. Kipman, A. Blake, Real-time human pose recognition in parts from  
753 single depth images, in: Machine Learning for Computer Vision, Vol.  
754 411, 2013, pp. 119–135.
- 755 [7] C.-S. Lee, A. M. Elgammal, Modeling view and posture manifolds for  
756 tracking, in: IEEE International Conference on Computer Vision, 2007,  
757 pp. 1–8.
- 758 [8] Y. M. Lui, Advances in matrix manifolds for computer vision, in: Image  
759 and Vision Computing, Vol. 30, 2012, pp. 380 – 388.

- 760 [9] M. T. Harandi, C. Sanderson, S. Shirazi, B. C. Lovell, Kernel analysis  
761 on grassmann manifolds for action recognition, in: Pattern Recognition  
762 Letters, Vol. 34, 2013, pp. 1906 – 1915.
- 763 [10] M. Bregonzio, T. Xiang, S. Gong, Fusing appearance and distribution  
764 information of interest points for action recognition, in: Pattern Recog-  
765 nition, Vol. 45, 2012, pp. 1220 – 1234.
- 766 [11] S. O’Hara, Y. M. Lui, B. A. Draper, Using a product manifold distance  
767 for unsupervised action recognition, in: Image and Vision Computing,  
768 Vol. 30, 2012, pp. 206 – 216.
- 769 [12] J. Aggarwal, M. Ryoo, Human activity analysis: A review, in: ACM  
770 Computing Surveys, Vol. 43, 2011, pp. 1–43.
- 771 [13] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods  
772 for action representation, segmentation and recognition, in: Computer  
773 Vision and Image Understanding, Vol. 115, 2011, pp. 224–241.
- 774 [14] A. Veeraraghavan, A. Roy-Chowdhury, R. Chellappa, Matching shape  
775 sequences in video with applications in human movement analysis,  
776 in: IEEE Transactions on Pattern Analysis and Machine Intelligence,  
777 Vol. 27, 2005, pp. 1896–1909.
- 778 [15] M. F. Abdelkader, W. Abd-Almageed, A. Srivastava, R. Chellappa,  
779 Silhouette-based gesture and action recognition via modeling trajecto-  
780 ries on riemannian shape manifolds, in: Computer Vision and Image  
781 Understanding, Vol. 115, 2011, pp. 439 – 455.

- 782 [16] D. Gong, G. Medioni, Dynamic manifold warping for view invariant  
783 action recognition, in: IEEE International Conference on Computer Vi-  
784 sion, Barcelona, Spain, 2011, pp. 571–578.
- 785 [17] D. Gong, G. Medioni, X. Zhao, Structured time series analysis for human  
786 action segmentation and recognition, in: IEEE Transactions on Pattern  
787 Analysis and Machine Intelligence, Vol. PP, 2014, pp. 1–1.
- 788 [18] P. Turaga, A. Veeraraghavan, A. Srivastava, R. Chellappa, Statistical  
789 computations on grassmann and stiefel manifolds for image and video-  
790 based recognition, in: IEEE Transactions on Pattern Analysis and Ma-  
791 chine Intelligence, Vol. 33, 2011, pp. 2273–2286.
- 792 [19] P. Turaga, R. Chellappa, Locally time-invariant models of human ac-  
793 tivities using trajectories on the grassmannian, in: IEEE Conference on  
794 Computer Vision and Pattern Recognition, 2009, pp. 2435–2441.
- 795 [20] K. Guo, P. Ishwar, J. Konrad, Action recognition in video by sparse  
796 representation on covariance manifolds of silhouette tunnels, in: Recog-  
797 nizing Patterns in Signals, Speech, Images and Videos, Vol. 6388, 2010,  
798 pp. 294–305.
- 799 [21] Y. M. Lui, J. R. Beveridge, Tangent bundle for human action recogni-  
800 tion, in: IEEE International Conference on Automatic Face and Gesture  
801 Recognition, 2011, pp. 97–102.
- 802 [22] Y. M. Lui, Tangent bundles on special manifolds for action recognition,  
803 in: IEEE Transactions on Circuits and Systems for Video Technology,  
804 Vol. 22, 2012, pp. 930–942.

- 805 [23] S. Shirazi, M. T. Har, C. S, A. Alavi, B. C. Lovell, Clustering on grass-  
806 mann manifolds via kernel embedding with application to action anal-  
807 ysis, in: International Conference on Image Processing, 2012, pp. 781–  
808 784.
- 809 [24] M. T. Harandi, C. Sanderson, S. Shirazi, B. C. Lovell, Kernel analysis  
810 on grassmann manifolds for action recognition, in: Pattern Recognition  
811 Letters, Vol. 34, 2013, pp. 1906 – 1915.
- 812 [25] J. Gall, A. Yao, L. Van Gool, 2D action recognition serves 3d human  
813 pose estimation, in: European Conference on Computer Vision, Vol.  
814 6313, 2010, pp. 425–438.
- 815 [26] L. Chen, H. Wei, J. Ferryman, A survey of human motion analysis using  
816 depth imagery, in: Pattern Recognition Letters, Vol. 34, 2013, pp. 1995  
817 – 2006.
- 818 [27] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, J. Gall, A survey on human  
819 motion analysis from depth data, in: Time-of-Flight and Depth Imaging.  
820 Sensors, Algorithms, and Applications, Vol. 8200, 2013, pp. 149–187.
- 821 [28] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D  
822 points, in: IEEE Conference on Computer Vision and Pattern Recogni-  
823 tion Workshops, 2010, pp. 9–14.
- 824 [29] B. Ni, G. Wang, P. Moulin, Rgbd-hudaact: A color-depth video database  
825 for human daily activity recognition, in: International Conference on  
826 Computer Vision Workshops, 2011, pp. 1147–1153.

- 827 [30] L. Xia, J. Aggarwal, Spatio-temporal depth cuboid similarity feature  
828 for activity recognition using depth camera, in: IEEE Conference on  
829 Computer Vision and Pattern Recognition, 2013, pp. 2834–2841.
- 830 [31] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, M. F. Campos,  
831 STOP: Space-time occupancy patterns for 3D action recognition from  
832 depth map sequences, in: Progress in Pattern Recognition, Image Anal-  
833 ysis, Computer Vision, and Applications, Vol. 7441, 2012, pp. 252–259.
- 834 [32] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3D action  
835 recognition with random occupancy patterns, in: European Conference  
836 on Computer Vision, 2012, pp. 872–885.
- 837 [33] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion  
838 maps-based histograms of oriented gradients, in: international confer-  
839 ence on ACM Multimedia, New York, NY, USA, 2012, pp. 1057–1060.
- 840 [34] O. Oreifej, Z. Liu, Hon4d: Histogram of oriented 4d normals for activity  
841 recognition from depth sequences, in: IEEE Conference on Computer  
842 Vision and Pattern Recognition, Washington, DC, USA, 2013, pp. 716–  
843 723.
- 844 [35] L. Xia, C.-C. Chen, J. K. Aggarwal, View invariant human action recog-  
845 nition using histograms of 3D joints, in: Computer Vision and Pattern  
846 Recognition Workshops, 2012, pp. 20–27.
- 847 [36] X. Yang, Y. Tian, Eigenjoints based action recognition using naive bayes  
848 nearest neighbor, in: Computer Vision and Pattern Recognition Work-  
849 shops, 2012, pp. 14–19.

- 850 [37] T. Giorgino, Computing and visualizing dynamic time warping align-  
851 ments in R: The dtw package, in: *Journal of Statistical Softwar*, Vol. 31,  
852 2009, p. 1–24.
- 853 [38] M. Reyes, G. Dominguez, S. Escalera, Featureweighting in dynamic  
854 timewarping for gesture recognition in depth data, in: *IEEE Interna-*  
855 *tional Conference on Computer Vision Workshops*, 2011, pp. 1182–1188.
- 856 [39] S. Sempena, N. Maulidevi, P. Aryan, Human action recognition using  
857 Dynamic Time Warping, in: *International Conference on Electrical En-*  
858 *gineering and Informatics*, 2011, pp. 1–5.
- 859 [40] M. Bautista, A. Hernandez-Vela, V. Ponce, X. Perez-Sala, X. Bar, O. Pu-  
860 jol, C. Angulo, S. Escalera, Probability-based dynamic time warping for  
861 gesture recognition on RGB-D data, in: *Advances in Depth Image Anal-*  
862 *ysis and Applications*, Vol. 7854, 2013, pp. 126–135.
- 863 [41] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. Laviola, Jr., R. Sukthankar,  
864 Exploring the trade-off between accuracy and observational latency in  
865 action recognition, in: *International Journal of Computer Vision*, Vol.  
866 101, 2013, pp. 420–436.
- 867 [42] M. Barnachon, S. Bouakaz, B. Boufama, E. Guillou, Ongoing human ac-  
868 tion recognition with motion capture, in: *Pattern Recognition*, Vol. 47,  
869 2014, pp. 238 – 247.
- 870 [43] M. Tenorth, J. Bandouch, M. Beetz, The TUM kitchen data set of every-  
871 day manipulation activities for motion tracking and action recognition,

- 872 in: International Conference on Computer Vision Workshops, 2009, pp.  
873 1089–1096.
- 874 [44] S. Azary, A. Savakis, A spatiotemporal descriptor based on radial dis-  
875 tances and 3D joint tracking for action classification, in: IEEE Interna-  
876 tional Conference on Image Processing, 2012, pp. 769–772.
- 877 [45] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action  
878 recognition with depth cameras, in: IEEE Conference on Computer  
879 Vision and Pattern Recognition, 2012, pp. 1290–1297.
- 880 [46] S. Althloothi, M. H. Mahoor, X. Zhang, R. M. Voyles, Human activity  
881 recognition using multi-features and multiple kernel learning, in: Pat-  
882 tern Recognition, Vol. 47, 2014, pp. 1800 – 1812.
- 883 [47] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi,  
884 A. Del Bimbo, Space-time pose representation for 3D human action  
885 recognition, in: Workshop on Social Behaviour Analysis ICIAP, Vol.  
886 8158, 2013, pp. 456–464.
- 887 [48] S. Azary, A. Savakis, Grassmannian sparse representations and motion  
888 depth surfaces for 3D action recognition, in: IEEE Conference on Com-  
889 puter Vision and Pattern Recognition Workshops, 2013, pp. 492–499.
- 890 [49] X. Zhang, Y. Yang, L. Jiao, F. Dong, Manifold-constrained coding and  
891 sparse representation for human action recognition, in: Pattern Recog-  
892 nition, Vol. 46, 2013, pp. 1819 – 1831.
- 893 [50] R. Li, P. Turaga, A. Srivastava, R. Chellappa, Differential geometric

- 894 representations and algorithms for some pattern recognition and com-  
895 puter vision problems, in: *Pattern Recognition Letters*, Vol. 43, 2014,  
896 pp. 3 – 16.
- 897 [51] H. Wang, C. Yuan, G. Luo, W. Hu, C. Sun, Action recognition using  
898 linear dynamic systems, in: *Pattern Recognition*, Vol. 46, 2013, pp. 1710  
899 – 1718.
- 900 [52] G. Doretto, A. Chiuso, Y. N. Wu, S. Soatto, Dynamic textures, in:  
901 *International Journal of Computer Vision*, Vol. 51, 2003, pp. 91–109.
- 902 [53] A. Bissacco, A. Chiuso, Y. Ma, S. Soatto, Recognition of human gaits,  
903 in: *IEEE Conference on Computer Vision and Pattern Recognition*,  
904 Vol. 2, 2001, pp. 52–57.
- 905 [54] A. Edelman, T. A. Arias, S. T. Smith, The geometry of algorithms with  
906 orthogonality constraints, in: *SIAM Journal on Matrix Analysis and*  
907 *Applications*, Vol. 20, 1998, pp. 303–353.
- 908 [55] A. Srivastava, E. Klassen, S. Joshi, I. Jermyn, Shape analysis of elastic  
909 curves in euclidean spaces, in: *IEEE Transactions on Pattern Analysis*  
910 *and Machine Intelligence*, Vol. 33, 2011, pp. 1415–1428.
- 911 [56] S. Kurtek, A. Srivastava, E. Klassen, Z. Ding, Statistical modeling of  
912 curves using shapes and related features, in: *Journal of the American*  
913 *Statistical Association*, Vol. 107, 2012, pp. 1152–1165.
- 914 [57] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines,  
915 in: *ACM Transactions on Intelligent Systems and Technology*, Vol. 2,  
916 2011, pp. 1–27.



- 917 [58] L. Xia, C.-C. Chen, J. K. Aggarwal, View invariant human action recog-  
918 nition using histograms of 3D joints., in: IEEE Conference on Computer  
919 Vision and Pattern Recognition Workshops, 2012, pp. 20–27.